DOCUMENT RESUME

ED 292 233 EC 202 015

AUTHOR Kotze, Paula
TITLE Informatics--Preparation for the Realities of the Future.
PUB DATE Jun 84
NOTE 28p.; Paper presented at the International Conference: Education for the Gifted "Ingenium 2000" (Stellenbosch, Republic of South Africa, June 26-29, 1984).
PUB TYPE Speeches/Conference Papers (150) -- Reports - Descriptive (141)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Computer Literacy; *Computer Uses in Education; *Course Descriptions; Elementary Secondary Education; Foreign Countries; *Gifted; *Problem Solving; *Programing; Special Schools
IDENTIFIERS South Africa

ABSTRACT

The paper describes the informatics curriculum (the study of computer hardware and software as a tool in problem solving) in a special school for gifted children in South Africa. The program's aims (including development of a structured approach to general problem solving and stimulation of pupil interest in technology) are listed and discussed. A syllabus covering a 6-year program from standard (grade) 5 through standard 10 is presented. Concepts covered include computer components, algorithms and flow charts, BASIC commands, number systems, subroutines, functions, Boolean algebra, logical operations, systems theory, operations on matrixes, FORTRAN, computer graphics, and PASCAL. Examples of students' written programs and subprograms and a sample problem for student solution are offered. (DB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☒ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED292233

INFORMATICS - PREPARATION FOR THE REALITIES OF THE FUTURE

CONFERENCE THEME: CURRICULUM AND METHODOLOGY

Mrs. Paula Kotzé
Department of Computer Science & Information Systems
University of South Africa
P.O. Box 392
Pretoria
0001
Republic of South Africa

This paper or a version close to it has been, or is being, offered for publication. The publisher's permission to reprint the paper in a publication on Ingenium 2000 will be supplied by the author.

P. KOTZÉ

EC202015

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
P. Kotzé
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

EXTRACURRICULAR CENTRE FOR HIGHLY GIFTED CHILDREN, ONDERWYSKOLLEGE, PRETORIA

INFORMATICS - PREPARATION FOR THE REALITIES OF THE FUTURE

by Mrs. Paula Kotzé, lecturer in the Department of Computer Science and Information Systems, University of South Africa.

INFORMATICS IS THE STUDY OF COMPUTER HARDWARE AND SOFTWARE AS AN AID TO SOLVING PROBLEMS IN EVERYDAY LIFE AS WELL AS IN THE FUTURE.

1. INTRODUCTION

In the Transvaal Education Department's framework of programmes for highly gifted children it is stated that the purpose of these programmes is not to pre-empt university courses, but to assist these pupils to gain valuable, useful and interesting knowledge which will have an enriching influence on their lives or promote further study in an intensive way.

The field of Informatics offers an excellent opportunity of achieving these objectives in a truly effective manner. Like Mathematics, Informatics is a generally formative subject: Whereas Mathematics is the language of all of Science, and as such is indispensable in the development of a capacity for logical and analytical thought, Informatics is the study of logical, discrete problem solving which has a contribution to make in all fields of study and societal structures. Even more important, however, is the fact that Informatics exercises a penetratingly formative influence on pupils' capacity to reason and their mode of thought.

The framework of programmes offered must be seen against this background. The general objective is that of providing structured guidance in the development of a capacity for creative problem solving, and subordinate to this is the presentation of detail

concerning the particular idiosyncracies of modern computers and computer languages.

In order to be successful, the programme must ensure the creation of a milieu which offers the pupil enough of a challenge, increases motivation and encourages ambition. The program has been planned to comply with these requirements. It should allow the highly gifted pupil's talents to unfurl abundantly, providing that he adapts these talents to the demands of Informatics.

2. THE PROGRAMME'S AIMS

2.1 The development of a structured approach to general problem solving.

2.2 The development of the pupil's capacity for creative thought.

2.3 Development of the quality of perseverance.

2.4 Systematic development of the pupil's interest in problem areas which extend beyond the merely personal and have a wider social significance.

2.5 Stimulation of the pupil's interest in the technological world of the future and the demands that man is going to make on that technology.

2.6 The advancement of the pupil's capacity for communication.

One of the most important characteristics of Informatics is that whilst permitting creative thought of the highest order, it also requires that the product of such thought be brought back to reality when it is implemented on a real computer with the necessity to comply with restricting language requirements and

technical hindrances. However, it is clearly stated that the programme will not only concentrate on the above-mentioned issues. Pupils will not be excluded or protected from the realities of the computer environment - on the contrary, it offers an opportunity of ensuring that the qualities of perseverance and self-discipline are promoted. In general, therefore, tasks should be completed in their entirety (although there are naturally some exceptions) and be accompanied by documentation - all of this in a way which contributes to the pupil's sense of self-esteem and represents real beacons of progress.

The program is aimed at attempting to ensure that the pupils will be capable of making their rightful contribution to society - obviously without losing sight of the pupil's own development or influencing him or her in a negative manner. In a balanced programme these aims will combine to form the desired whole.

As regards the pupil's capacity to communicate, both verbal and non-verbal communication must be encouraged at an early stage in order to ensure that he or she will eventually be able to make a real contribution to science and to society. Informatics lends itself to this very well, since everyone tackles and completes problems in a different way and pupils can compare notes on alternative methods of solving them. We also encourage the principle of working together, whereby two or more pupils put their heads together to solve a problem. Since the pupils often have different interests, they can consult one another for advice when their knowledge of a particular topic is inadequate. Since we sometimes literally throw pupils in at the deep end and they then have to acquire knowledge of a topic on their own, they gain a better idea of where they can obtain help in order to realize their own ideals in the process of looking up references in the library or elsewhere, and in consulting people with a broader knowledge.

The development of the pupil's own ideals is never lost sight of.

Highly gifted pupils' interpersonal communication often leaves a great deal to be desired. Since they often feel "odd man out" in an average group, they tend to become withdrawn. They often become impatient and feel that they are wasting their time since they immediately grasp and accept those things which constitute a problem to the average child. At the centre they communicate with equals and, since they often have to solve very difficult problems, they sometimes struggle too. This has a salutary effect on their relations with other people. As one pupil quite rightly said: "I now understand how children who are struggling with their work at school must feel". Communication with the average child thus improves. Some pupils are very antisocial when they come to the centre, but since our classes are small and pupils are given individual attention, they thaw sooner or later.

One fear which often plagues highly gifted children is the fear of making mistakes. Their parents and teachers often set such high targets that the children are afraid of blundering. Infor= matics is pre-eminently suited to sorting out this kind of mis= understanding, since it lends itself to making mistakes, but we point out that there is no shame attached to this and that one can learn by one's errors. No-one will be angry if a mistake is made - this will merely serve as an incentive to discover the correct solution. In other words we are already laying the foundations for research at school level.

3. PROGRAMME

3.1 INTRODUCTION

- There are no preconceived, rigid rules for the presentation of the topics.
- The person responsible for instruction must at all times bear in mind the framework of reference and level of readiness of the pupils concerned and adapt the programme accordingly.
- The classes consist of informal classical teaching as well as the individual and group work which make up the major part of the programme.

3.2 INFORMATICS SYLLABUS, 1984

N.B. The different modules follow on from each other and cannot be taken as separate units. The pupil must thus start at I regardless of his current school standard.

INFORMATICS I - Standard 5

I. Background:
   1. Historical development of computers
   2. Components of the computer
   3. Problem solving - algorithms and flow charts

II. Applications - Basic:
   1. Rem
   2. End
   3. Print, Print using
   4. Read, Restore, Data
   5. Input

6. Arithmetic expressions
7. If-then-else
8. Goto
9. For-next
10. While-wend
11. Elementary library functions

INFORMATICS II - Standard 6

I. Background:

1. Number systems
2. Problem solving - algorithms and flow charts

II. Applications - Basic

1. Revise: a) If-then-else
   b) For-next
   c) While-wend
2. Functions
3. Subroutines: a) Gosub
   b) Return
   c) End
4. Handling character strings

INFORMATICS III - Standard 7

I. Background:

1. Operations on vectors and matrices (dealing only with addition of rows, columns and diagonals)
2. Boolean Algebra: a) Logical expressions
   b) Meaning of OR, AND and NOT
   c) Truth tables for OR, AND and NOT
   d) Physical implementation of OR, AND and NOT

3. File structure: a) Sequential
b) Direct access

II. Applications - Basic

1. Vectors and matrices
2. Logical operations using AND and OR
3. Sequential files
4. Direct access files

INFORMATICS IV - Standard 8

I. Background:

a) Systems theory
b) Operations on matrices: Addition, scalar multiplication, product of two matrices
c) Introduction to computer graphics

II. Applications:

a) Basic and FORTRAN
b) Computer graphics
c) Program packages such as Multiplan and Visicalc
d) The writing of a small system

INFORMATICS V - Standard 9

I. Background:

1. Data structures

a) Unrestricted lists
i) Ordinary arrays
ii) Linked lists
b) Restricted lists
i) Stacks

ii) Queues
iii) Deques
iv) Trees

2. Computer graphics

II. Applications

1. Pascal
2. Computer graphics
3. Use of the word processor and other program packages

INFORMATICS VI - Standard 10

I. Background:

1. Introduction to the analysis, design and implementation of information systems
2. Computer graphics

II. Applications in the language of the pupil's choice

3.3 THE LEARNING OF COMPUTER LANGUAGES

The first computer language the pupil learns is an extended, structured Basic. The language is specified for them in detail, with the necessary examples showing them how to use the various control structures.

The learning of a second language is relatively easy. We use the comparative principle, showing what certain coding in the language already learned looks like, and thereafter what it looks like in the new language. The same method is followed when it comes to the third and fourth languages. If the new languages contain topics not covered in the language already known, these are dis= cussed in detail.

The reason for beginning with Basic is that most of the pupils acquire their own microcomputers on which they can use Basic. At the moment the language they learn next is FORTRAN, the reason being that it is at present the first high-level language the pupils learn in Computer Studies. The pupils taking both Informatics and Computer Studies can thus do their applications for Computer Studies in the Informatics class, where they acquire "hands-on" experience, which is not the case in Computer Studies. After FORTRAN they proceed with Pascal which is the language the universities are going in for at the moment. We also envisage teaching COBOL to our first INFORMATICS VI group since it is the number one language for commercial data-processing at present. The pupils will thus be very well equipped for using the computer to solve problems in any field of study.

4. EXAMPLES OF THE PUPIL'S WORK

4.1 THE HANDLING OF CHARACTER STRINGS

The program which follows was written by a Standard 8 pupil with an Informatics I background. At the time the program was written he had been attending the centre for three months and had spent 24 hours on Informatics, of which only 4 hours was devoted to theoretical work. He was assigned a vacation project for which he had to study the handling of character strings and then write a program to convert numbers in Roman numerals into decimal equivalents.

```
5 REM   ************************************************************
10 REM   PROGRAM OM ROMEINSE SYFERS NA DESIMALE SYFERS OM TE SKAKEL
20 REM  ************************************************************
30 I% = DEFLPRINT("[SPL]")
40 PRINT "ROMEINS","DESIMAAL"
50 PRINT "-------","-------"
60 INPUT ROM$
70 PRINT ROM$
80 ALFA$ = "CLXVI"
90 LENGTE = LEN(ROM$)
100 DIM ANTW(LENGTE)
120 FOR I = 1 TO 5
130     READ DES
140     ALFACHR$ = MID$(ALFA$,I,1)
150      FOR J = 1 TO LENGTE
160          DESCHR$ = MID$(ROM$,J,1)
170          IF ALFACHR$ = DESCHR$
                THEN ANTW(J) = DES
180      NEXT J
190 NEXT I
200 FOR K = 1 TO LENGTE-1
```

```
210       IF ANTW(K) < ANTW(K+1)
             THEN GETAL1 = GETAL1 + ANTW(K)
             ELSE GETAL2 = GETAL2 + ANTW(K)
220 NEXT K
230 GETAL2 = GETAL2 + ANTW(LENGTE)
240 GETAL =GETAL2 - GETAL1
250 PRINT GETAL
260 I% =DEFLPRINT("[SPL]")
265 DATA 100,50,10,5,1
270 END
```

4.2 USE OF SUBPROGRAMS

The following program was written by a Standard 8 pupil with Informatics I and II as background. His assignment was to write a program for a game called Binopoly, using subprograms. Binopoly is similar to Monopoly, except that the playing board is limited to 20 positions costing R1 each. The fine for landing on another player's property is R1. Each player is given an initial amount of R10 to spend. The game ends as soon as any player goes bankrupt.

```
5 I% = DEFLPRINT("[SPL]")
10 RANDOMIZE
20 REM ****** BINOPOLY PROGRAM ******
30 REM ******************************
40 DIM DOBBEL$(7,7), NAAM$(4), STUK$(4), POSBORD$(8,61), POSISIE$(10)
50 DIM GELD(4), BESIT(20), PLEK(4)
60 DATA 0,1,2,3,4,5,6,7,8,9
70 FOR I = 1 TO 10
80     READ POSISIE$(I)
90 NEXT I
100 LPRINT "Hierdie is n spel genaamd BINOPOLY.  Dit is soortgelyk
aan Monopoly,"
110 LPRINT "behalwe dat die bord beperk is tot 20 posisies wat elk
R1 kos.  Die"
120 LPRINT "boete as n speler op n ander speler se eiendom land is
R1.  Elke"
130 LPRINT "speler kry n aanvanklike bedrag van R10 om te spandeer.
Die spel"
140 LPRINT "eindig as die eerste speler bankrot raak.  Die spele=
tjie is nie"
150 LPRINT "interaktief nie."
160 LPRINT
170 GOSUB 550
```

```
180 GOSUB 1640
190 GOSUB 1310
200 RONDTE = 1
210 EINDE = 0
220 WHILE EINDE = 0
230       FOR I = 1 TO 4
240          LPRINT "RONDTE ";RONDTE : LPRINT "----------"
250          LPRINT NAAM$(I);";" JY HET 'N "
260          II = I
270          GOSUB 1420
280          I = II
290          PLEK(I) = PLEK(I) + D
300          IF PLEK(I) > 20 THEN PLEK(I) = PLEK(I) - 20
310          K = BESIT(PLEK(I))
320          IF K <> I
                THEN IF K <> 0
                        THEN GELD(K) = GELD(K)+1 :GELD(I) = GELD(I) - 1
                        ELSE BESIT(PLEK(I))=I:GELD(I)=GELD(I)-1
330          IF GELD(I) = 0 THEN EINDE = 1
340       NEXT 1
350       RONDTE = RONDTE + 1
360 REM DRUK HUIDIGE POSISIES
370 GOSUB 980
380 FOR I = 4 TO 7
390     FOR J = 2 TO 59 STEP 3
400         POSBORD$(I,J) = " "
410         IF J = 3 * PLEK(I-3) -1
               THEN POSBORD$(I,J) = STUK$(I-3)
420     NEXT J
430 NEXT I
440 LPRINT "Die huidige stand op die bord is"
450 GOSUB 910
460 GOSUB 1310
470 WEND
```

```
480 WENNER = 1
490 GROOT = GELD(1)
500 FOR I = 2 TO 4
510     IF GROOT < GELD(I)
           THEN WENNER = I: GROOT = GELD(I)
520 NEXT I
530 LPRINT NAAM$(WENNER);" IS DIE WENNER"
531 I% = DEFLPRINT("[SPL]")
540 STOP
550 REM ****** SUBROETINE: BORD ******
560 LPRINT "DIE SPEELBORD LYK AS VOLG"
570 LPRINT
580 LPRINT "************************"
590 LPRINT "*1  *2  *3  *4  *5  *6  *"
600 LPRINT "*   *   *   *   *   *   *"
610 LPRINT "************************"
620 LPRINT "*20 *              *5  *"
630 LPRINT "*   *              *   *"
640 LPRINT "*****              *****"
650 LPRINT "*19 *              *6  *"
660 LPRINT "*   *              *   *"
670 LPRINT "*****              *****"
680 LPRINT "*18 *              *7  *"
690 LPRINT "*   *              *   *"
700 LPRINT "*****              *****"
710 LPRINT "*17 *              *8  *"
720 LPRINT "*   *              *   *"
730 LPRINT "*****              *****"
740 LPRINT "*16 *              *9  *"
750 LPRINT "*   *              *   *"
760 LPRINT "************************"
770 LPRINT "*15 *14 *13 *12 *11 *10 *"
780 LPRINT "*   *   *   *   *   *   *"
790 LPRINT "************************"
```

```
800 LPRINT
810 LPRINT "POSISIE 1 IS DIE BEGIN"
820 LPRINT "DAAR WORD KLOKSGEWYS GESPEEL"
830 LPRINT
840 LPRINT "DIE STAND VAN SPELERS OP DIE BORD SAL AS VOLG GETOON
          WORD."
850 LPRINT "ALLE SPELERS IS AANVANKLIK IN POSISIE 1."
860 LPRINT
870 GOSUB 980
880 FOR I = 4 TO 7
890     POSBORD$(1,2) = STUK$(1-3)
900 NEXT I
910 FOR I = 1 TO 8
920     FOR J = 1 TO 61
930         LPRINT POSBORD$(I,J);
940     NEXT J
950       LPRINT
960 NEXT I
970 RETURN
980 REM ****** SUBROETINE: POSISIE OP BORD
990 FOR I = 1 TO 8
1000     FOR J = 1 TO 61
1010         POSBORD$(I,J) = " "
1020     NEXT J
1030 NEXT I
1040 FOR I = 1 TO 8
1050     FOR J = 1 TO 61 STEP 3
1060         POSBORD$(I,J) = "*"
1070     NEXT J
1080 NEXT I
1090 FOR J = 2 TO 59 STEP 3
1100     POSBORD$(1,J) = "*"
1110    POSBORD$(1,J+1) = "*"
1120    POSBORD$(3,J) = "*"
```

```
1130     POSBORD$(3,J+1) = "*"
1140     POSBORD$(8,J) = "*"
1150     POSBORD$(8,J+1) = "*"
1160 NEXT J
1170 I = 2
1180 FOR J = 2 TO 26 STEP 3
1190     POSBORD$(2,J) = POSISIE$(I)
1200     I = I + 1
1210 NEXT J
1220 I = 1
1230 FOR J = 29 TO 56 STEP 3
1240     POSBORD$(2,J) = "1"
1250     POSBORD$(2,J+1) = POSISIE$(I)
1260    I = I + 1
1270 NEXT J
1280 POSBORD$(2,59) = "2"
1290 POSBORD$(2,60) = "0"
1300 RETURN
1310 REM ****** SUBROETINE: GELD
1320 FOR I = 1 TO 4
1330 LPRINT NAAM$(I); " HET R"; GELD(I); "EN BESIT DIE VOLGENDE
          STANDPLASE:"
1340     TEL = 0
1350     FOR J = 1 TO 20
1360         IF BESIT(J) = I
                THEN LPRINT J: TEL = I
1370     NEXT J
1380     IF TEL = 0 THEN LPRINT "GEEN"
1390 LPRINT
1400 NEXT I
1410 RETURN
1420 REM ****** SUBROETINE: DOBBELSTEEN
1430 FOR I = 1 TO 7
```

```
1440     FOR J = 1 TO 7
1450         DOBBEL$(I,J) = " "
1460     NEXT J
1470 NEXT I
1480 D = INT(RND(1)*6)+1
1490 IF D = 1 OR D = 2 OR D = 5
        THEN DOBBEL$(4,4) = "*"
1500 IF D <> 1
        THEN DOBBEL$(2,2) = "*":DOBBEL$(6,6) = "*"
1510 IF D > 3
        THEN DOBBEL$(6,2) = "*": DOBBEL$(2,6) = "*"
1520 IF D = 6
        THEN DOBBEL$(4,2) = "*": DOBBEL$(4,6) = "*"
1530 LPRINT "-------"
1540 FOR I = 1 TO 7
1550     LPRINT "!";
1560     FOR J = 1 TO 7
1570         LPRINT DOBBEL$(I,J);
1580         DOBBEL$(I,J) = " "
1590     NEXT J
1600     LPRINT "!"
1610 NEXT I
1620 LPRINT "-------"
1630 RETURN
1640 REM ****** SUBROETINE: NAME, SPEELSTUKKE EN GELD
1650 DATA @,_,&,%
1660 FOR I = 1 TO 4
1670     READ STUK$(I)
1680     PRINT "WAT IS DIE NAAM VAN SPELER";I
1690     INPUT NAAM$(I)
1700     LPRINT NAAM$(I) ; ": JOU SPEELSTUK IS 'N ",STUK$(I)
1710     GELD(I) = 10
```

```
1720     PLEK(I) = 1
1730 NEXT I
1740 RETURN
1750 END
```

4.3 FIRST ACQUAINTANCE WITH PASCAL

The pupil who wrote the following program has experience of Informatics I, II and III and after the first lecture comparing Basic with Pascal, came up with the following program for a bubble sort.

```
program BORREL(input,output);
label FINISH;
var X:array [1..10] of integer;
    N,PASS,J,HOLD : integer;
    INTERCHANGE : boolean;
begin
     INTERCHANGE := true;
     for J := 1 to 10 do
     begin
          readln(X[J]);
     end;
     for PASS := 1 to 9 do
     begin
          if INTERCHANGE then
          begin
               INTERCHANGE := false;
               for J := 1 to 10 - PASS do
               begin
                    if X[J] > X[J+1] then
                    begin
                         INTERCHANGE := true;
                         HOLD := X[J];
                         X[J] :  X[J+1];
                         X[J+1] : HOLD;
                    end
                    else goto FINISH;
               end;
```

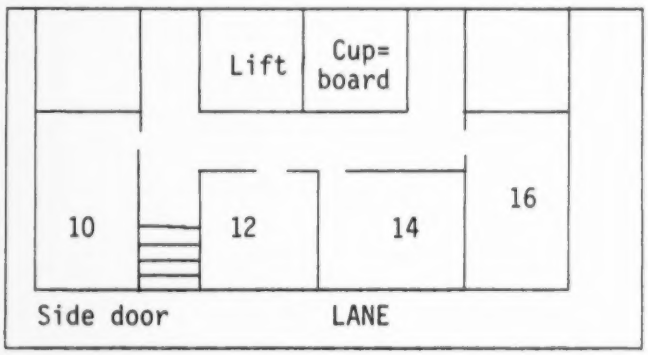
```
          end;
     end;
FINISH: for J := 1 to 10 do
        begin
             writeln('Gesorteerde lys');
             writeln(X[J]);
        end;
end.
```

4.4 AFTER DISCUSSION OF TOPICS WHICH ONLY OCCUR IN PASCAL

This assignment was entitled "MURDER AT THE METROLOPITAN CLUB". The pupils received the following clues and then had to determine who had committed the murder.

An initial enquiry by Scotland Yard yielded four suspects. The four were registered in adjoining rooms above the club. One of them was Sir Raymond Jasper, a noted member of the bar, and another an accountant by the name of Robert Holman. The other two, a Colonel Reginald Woodley and Mr James Pope, were visitors from Northumberland, unconnected with each other. The police assembled a list of facts concerning the crime and these four gentlemen.

Here is a partial floor plan of the club's boarding rooms and the list of clues assembled:



The clues are as follows:

1. Sir Raymond Jasper occupied Room 10.
2. The man occupying Room 14 had black hair.
3. Either Colonel Woodley or Sir Raymond wore a pince-nez.
4. Mr Pope always carried a gold pocket-watch.
5. One of the suspects was seen driving a four-wheel carraige.

6. The man with the pince-nez had brown hair.
7. Mr Holman wore a ruby signet ring.
8. The man in Room 16 had tattered cuffs.
9. Mr Holman occupied Room 12.
10. The man with the tattered cuffs had red hair.
11. The man in Room 12 had grey hair.
12. The man with a gold packet-watch occupied Room 14.
13. Colonel Woodley occupied a corner room.
14. The murderer had brown hair.

The following are two possible solutions in Pascal.

```
program METROPOLITANCLUB(OUTPUT);
const
     UNKNOWN = 0;
     RED = 1;        BLACK = 2;        GREY = 2;        BROWN = 4;
     PINCENEZ = 1;   GOLDWATCH = 2;    RUBYRING = 3;    TATTEREDCUFFS = 4;
     COLWOODLEY = 1; MRHOLMAN = 2;     MRPOPE = 3 ;     SIRRAYMOND = 4;
var
     SUSPECT, MURDERER: INTEGER;
     HAIR:   array [COLWOODLEY .. SIRRAYMOND] of integer;
     ATTIRE: array [COLWOODLEY .. SIRRAYMOND] of integer;
     ROOM:   array [COLWOODLEY .. SIRRAYMOND] of integer;
begin
  MURDERER := UNKNOWN;
  for SUSPECT := COLWOODLEY to SIRRAYMOND do begin
      HAIR[SUSPECT]   := UNKNOWN;
      ATTIRE[SUSPECT] := UNKNOWN;
      ROOM[SUSPECT]   := UNKNOWN
  end;
   ROOM[SIRRAYMOND] := 10;
   ATTIRE[MRPOPE]    := GOLDWATCH;
   ATTIRE[MRHOLMAN] := RUBYRING;
   ROOM[MRHOLMAN]   := 12;
```

```
(* Repeatedly try the remaining clues *)
SUSPECT := COLWOODLEY;
while MURDERER = UNKNOWN do begin
   if (ROOM[SUSPECT] = 14) then
      HAIR[SUSPECT] := BLACK;
   if (ATTIRE[SIRRAYMOND]<>UNKNOWN) and (ATTIRE[SIRRAYMOND]<>
      PINCENEZ) then ATTIRE[COLWOODLEY] := PINCENEZ;
   if (ATTIRE[COLWOODLEY]<>UNKNOWN) and (ATTIRE[COLWOODLEY]<>
      PINCENEZ) then ATTIRE[SIRRAYMOND] := PINCENEZ;
   if (ATTIRE[SUSPECT] = PINCENEZ) then
      HAIR[SUSPECT] := BROWN;
   if (ATTIRE[SUSPECT] = TATTEREDCUFFS) then
      HAIR[SUSPECT] := RED;
   if (ROOM[SUSPECT] = 16) then
      ATTIRE[SUSPECT]:= TATTEREDCUFFS;
   if (ROOM[SUSPECT]= 12) then
      HAIR[SUSPECT]:= GREY;
   if (ATTIRE[SUSPECT]= GOLDWATCH) then
      ROOM[SUSPECT] := 14;
   if (ROOM[SUSPECT] = 10) and (SUSPECT <> COLWOODLEY) then
      ROOM[COLWOODLEY] := 16;
   if (ROOM[SUSPECT] = 16) and (SUSPECT <> COLWOODLEY) then
      ROOM[COLWOODLEY] := 10;
   if (HAIR[SUSPECT] = BROWN) then
      MURDERER := SUSPECT;
   if (SUSPECT = SIRRAYMOND) then
      SUSPECT := COLWOODLEY
   else
      SUSPECT := SUSPECT + 1
end;
```

```
if MURDERER = COLWOODLEY then
      WRITE ('THE MURDERER IS COLONEL WOODLEY;')
else if MURDERER = MRHOLMAN then
      WRITE ('THE MURDERER IS MR. HOLMAN.')
else if MURDERER = MRPOPE then
      WRITE ('THE MURDERER IS MR. POPE.')
else
      WRITE ('THE MURDERER IS SIR RAYMOND.')
end.
```

OPTION 2

```
PROGRAM MURDER (output);
TYPE persons=(woodley, raymond, pope, holman, dont_know);
VAR hair : ARRAY[persons] of (black,brown,red,grey,unknown);
    attire : ARRAY [persons] of (pincenez,tattered_cuffs,gold_
             watch, ruby_ring,knownot);
    room : ARRAY [persons] of 9..16;
    suspect,murderer : persons;

BEGIN
    for suspect:=woodley to holman do
      BEGIN
         hair[suspect]:=unknown;
         attire[suspect]:=knownot;
         room[suspect]:=9;
      END;
   murderer:=dont_know;
   room[raymond]:=10;
   attire[pope]:=gold_watch;
   attire[holman]:=ruby_ring;
   room[holman]:=12;
   while murderer=dont_know do
      for suspect:=woodley to holman do
```

```
            BEGIN
               if room[suspect]=14 then hair[suspect]:=black;
               if (attire[raymond]<>pincenez) and (attire[raymond]<>
                   knownot) then attire[woodley]:=pincenez;
               if (attire[woodley]<>pincenez) and (attire[woodley]<>
                   knownot) then attire[raymond]:=pincenez;
               if attire[suspect]=pincenez then hair[suspect]:=brown;
               if attire[suspect]=tattered_cuffs then hair[suspect]:=red;
               if room[suspect]=16 then attire[suspect]:=tattered_cuffs;
               if room[suspect]=12 then hair[suspect]:=grey;
               if attire[suspect]=gold_watch then room[suspect]:=14;
               if (room[suspect]=10) and (suspect<>woodley) then
                  room[woodley]:=16;
               if (room[suspect]=16) and (suspect<>woodley) then
                  room[woodley]:=10;
               if hair[suspect]=brown then murderer:=suspect
            END;
      case murderer of
        woodley : writeln('Colonel Woodley.');
        raymond : writeln('Sir Raymond.');
        pope    : writeln('Mr Pope.');
        holman  : writeln("Mr Holman.');
    END
END.
```

5. BIBLIOGRAPHY

5.1 Peter Brown, Pascal from BASIC, Addison-Wesley, Philippines, 1982.

5.2 Henry Ledgard and Andrew Singer, Elementary Pascal, SRA, Chicago, 1982.

5.3 Programkomitee vir Hoogbegaafde Leerlinge, Voorgestelde Riglyne vir die Program vir Hoogbegaafde Leerlinge vir die Rigting Informatika-studie, Transvaalse Onderwysdeparte= ment, 1981.